International Journal of Computer (IJC)
ISSN 2307-4523 (Print & Online)

http://ijcjournal.org/

Classification of the Stance in Online Debates Using the Dependency Relations Feature

Thiri Kyaw[a]*, Sint Sint Aung[b]

[a]*University of Computer Studies, Mandalay (UCSM), Mandalay, Myanmar*
[b]*University of Computer Studies, Mandalay (UCSM), Mandalay, Myanmar*
[a]*thirikyaw@ucsm.edu.mm*
[b]*sintsintaung@ucsm.edu.mm*

Abstract

Online discussion forums offer Internet users a medium for discussions about current political debates. The debate is a system of claims regarding interactivity and representation. Users make claims to support their position in an online discussion with superior content. Factual accuracy and emotional appeal are critical attributes used to convince readers. A key challenge in debate forums is to identify the participants' stance, each of which is inter-dependent and inter-connected. The proposed system takes the post's linguistic features as input and outputs predictions for each post's stance label. Three types of features including Lexical, Dependency, and Morphology are used to detect the post's stance. Lexical features such as cue words are employed as surface features, and deep features include dependency and morphology features. Multinomial Naïve Bayes classifier is used to build a model for classifying stance and the Chi-Square method is used to select the good feature set. The performance of the stance classification system is evaluated in terms of accuracy. By analyzing the surface and deep features capturing the content of a post, the result of stance labels for this proposed system represents as *for* and *against*.

Keywords: stance classification; machine learning; sentiment analysis; opinion mining;

1. Introduction

In recent years, Internet users spend more time on social media sites than any other type of site, creating, sharing, and exchanging information and ideas in the form of text, image, video, etc. Using data from the most popular social media platforms, such as Twitter, Facebook, Instagram, Forums, or Web blogs, enables researchers from different fields to address questions about individual and group social opinions and behavior. Examples of the use of data from social media in research include, but are not limited to sentiment analysis, detection of mental disorders, stance classification, and rumor detection. Online debate sites are one of the social networks where users can take a stance and argue in support or opposition of the debate topics. Different personal opinions on the web are, from the analysis point of view, a valuable great resource of user-generated data and learning user strategies in persuading readers how to support their stances. Users' posts can have a mixed set of emotions in which some sentences might support one topic and others might oppose that same topic. An author may participate in multiple discussions on the same topics and discuss multiple topics. Users debating sites freely their opinions, using informal and social language, providing a diverse and much harder environment for predicting their stances. The automated identification of stance has common applications in information retrieval, text summarization, recommendation systems, targeted advertising, political polling, and product reviews. We propose the efficient surface and deep features for the stance classification system because difficult to predict the stance using a traditional classification model which only considers surface features.

2. Related Works

Stance classification is to determine the stance of a post written for a two-sided topic discussed in an online debate forum (i.e., *for* or *against*). Previous work has focused on (1) congressional floor debates, (2) company-internal discussions, (3) online forums ideological debates, and (4) hot-event oriented debates on social media. Somasundaran and Wiebe [1, 2] find opinion-target pair expressions as associations among opinion/polarity, targets, and topics. Firstly, they [1] identify the opinion word, finding into the subjectivity lexicon, and then replacing the word with its polarity. Secondly, they use syntactic rules for finding targets of opinions. They [2] build a supervised system using sentiment lexicon and arguing opinions and their targets as features. To extract the arguing trigger expression, they use unigrams, bigrams, and trigrams starting at the first word from the MPQA corpus to create an arguing lexicon. Modal verbs and syntactic rules append to these features. They prove sentiment and arguing expressions of opinions are useful for debate side stance classification. Anand and his colleagues [3] focus on rebuttal and stance classification using Naïve Bayes and JRip classifiers with contextual features of the parent post. For particular topics, they showed that the parent post features are better than without any contextual features. Walker and his colleagues [4] partition the dialogic relations of rebuttal links and the same author links by using lexical features and parent-post context features, and the MaxCut algorithm that assigns the debate stance without considering based on the individual post's stance but based on the whole partition. Hasan and Ng [5] propose stance-supported sentence-level reason classification with N-gram, quotation, frame-semantic, dependency-based, and positional features. They examine different ways of modeling computed stance information to improve reason classification experiments on their reason-annotated corpus of ideological debate posts from four popular domains. Work by [6] on social and political issues predict not only the stance label for each post using both linguistic features of the post and linear SVM as the local classifier but also the stance relation between authors and posts using Probabilistic Soft Logic (PSL) model. Reference [7] calculate the reaction coefficients based on the results of the agree/disagree/neutral classes from reply-to pairs activities and opinion expressions in the textual contents and then identify participants' positions in online debate, support or oppose, using BiqMac. Reference [8] use Anand et al.'s approach as the first baseline model and incorporating author constraints into it as the second baseline model. They employ two types of extra-linguistic constraints, user-interaction constraints (UCs) and ideological constraints (ICs) to classify stance labels of debate posts into the second baseline model. Sobhani and his colleagues [9] propose argument mining and stance classification of online news comments. They applied Non-Negative Factorization to extract topics for argument tagging. They used a linear SVM with TF-IDF as features and the predicted argument tags as additional features for stance classification. Reference [10] identify the attitudes, agreement and disagreement of users in online discussion based on isotonic Conditional Random Fields. Their predictions are made on the sentence or segment level. They construct a sentiment lexicon by using lexical features, discourse features, syntactic/semantic features, conversation features, and sentiment features. On two existing online discussion corpora, the Authority and Alignment in Wikipedia a Discussions (AAWD) corpus and the Internet Argument Corpus (IAC), they evaluate their system.

3. Stance Classification System

We introduce our proposed system, which includes five different modules: (1) data gathering module, (2) linguistic pre-processing module, (3) feature generation module, (4) feature selection module, and (5) stance classification module. The proposed system is shown in Fig 1.

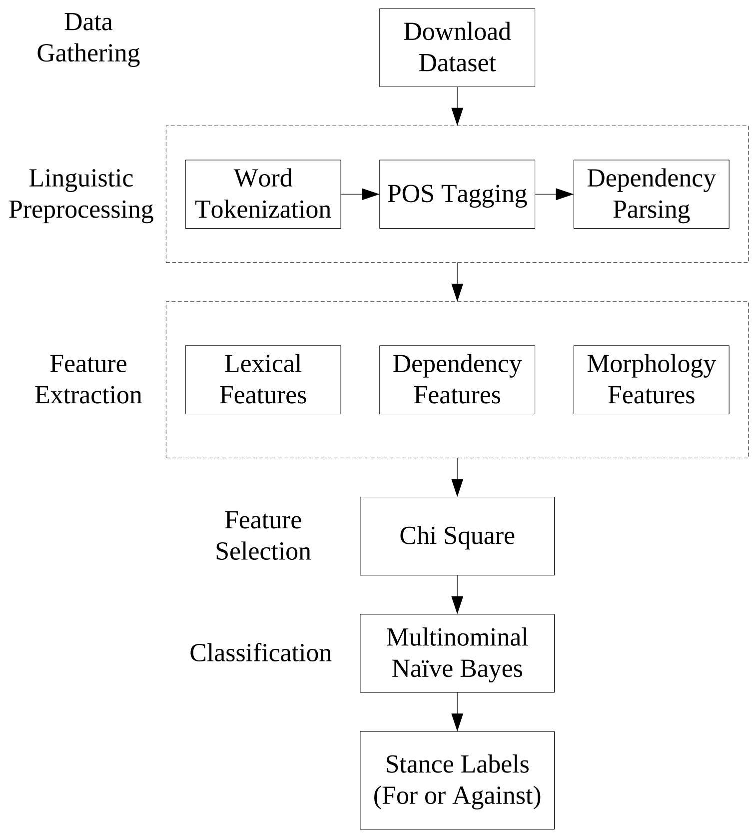

Fig. 1. The architecture of the stance classification system

3.1. Data gathering module

We used the ideological debates in public forums dataset which has monologic posts about six popular issues: Abortion rights, Creation, Gay rights, Existence of God, Gun rights, and Healthcare created by Somasundaran and Wiebe [2]. There are a total of 7128 posts including 1150, 1229, 2063, 952, 1068, and 666 posts from an online debate forum for the six domains. A debate has a topic (e.g., “Abortion should be banned?”) about which numerous authors wrote several posts. Each debate post has three parts: domain-level stance, original topic, and original stance text.

3.2. Linguistic pre-processing module

The linguistic pre-processing module is used to view the text documents in a simple word format which can be processed more quickly and efficiently. For the pre-processing module, we utilized the StanfordCoreNLP dependency parser [16].

Word Tokenization: A post is treated as a string, and then partitioned into a list of tokens.

POS Tagging: A part-of-speech tagger (POS Tagger) assigns to each word (and another token) part of the speech, such as nouns, verbs, adjectives, etc.

Dependency Parsing: Dependency parsing (DP) is a word centric parse that builds named, ordered relations between pairs of words in a sentence. The two words at either end of a relation are called respectively the head and the dependent.

3.3. Feature generation module

Post content is represented as a vector where the following feature represents the presence or absence of the features in the post. Generate the surface and deep features by using Stanford Dependency parser. There are three types of features: (1) lexical features, (2) dependency features, and (3) morphology features. Lexical features are used as surface features, and the dependency and morphology features are used as deep features.

(1) Lexical Features

Lexical features can capture the surface representation of each post. The first feature set consists of the following features collected from the training posts which encoded them as binary features that indicated the occurrence of a given post. The feature of the discourse cues captures binary representation for the post's few words that often contain discourse cues. Walker et al., [11] developed the Internet Argument Corpus (IAC) for stance classification. They constructed a list of discourse markers; in a quote response, 17 of these occurred at least 50 times. The top disagreement markers were ***really***, ***no***, ***actually***, ***but***, ***so*** and ***you mean***. The most agreeable markers were ***yes***, ***I know***, ***I believe***, ***I think*** and ***just***. Six disagreement markers and five agreeable markers were used for discourse cue words in the lexical features' extraction method. Figure 2 is a procedure for extraction the lexical features. For each grammatical relation, we can examine the dependency relation (reln), governor word (gov), and dependent word (dep). We can consider the unigram cue word and then the bigram cue word. Unigram cue words are ***actually***, ***but***, ***really***, ***no***, ***so***, ***just***, and ***yes*** and bigram cue words are ***believe***, ***know***, and ***think***. For unigram cue word, we can consider gov to be equal unigram cue word or dep to be equal unigram cue word, and we can take this grammatical relation as lexical features. For bigram cue word, we can find reln as equal nsubj and gov as equal bigram cue word or ***mean*** and dep as equal ***i*** or ***you***, and we can capture this grammatical relation as lexical features. This system also extracts the lexical features relations by using only dependency relations [15]. There are 29 different types of grammatical relations. Using feature selection, only 11 relations are preserved.

```
Algorithm 1: Lexical Features Extraction Method
Input   debateCorpus, cueWords1 {actually, but, really, no, so, just, yes}, cueWords2{believe, know, think}
Output  lexicalFeatures
        for i in size(debateCorpus) do
            sentences ← sentenceAnnotation(debateCorpus(i))
            for j in size(sentences) do
                dependencies ← dependencyAnnotation(sentences(j))
                for k in size(dependencies) do
                    reln ← relation of dependencies(k)
                    gov ← governor of dependencies(k)
                    dep ← dependent of dependencies(k)
                    if gov = cueWords1 or dep = cueWords1 then
                        lexicalFeatures ← dependencies(k)
                    elseif reln = nsubj and gov = cueWords2 and dep = "i" then
                        lexicalFeatures ← dependencies(k)
                    elseif reln = nsubj and gov = "mean" and dep = "you" then
                        lexicalFeatures ← dependencies(k)
                    else
                        not lexical features
                    end if
                end for
            end for
        end for
```

Fig. 2. Lexical features extraction method

(2) Dependency Features

The dependency parse for a given sentence is a set of triples, composed of a grammatical relation holds (rel_i, w_j, w_k), where rel_i is the dependency relation between w_j and w_k. The word w_j is usually referred to as the head word in the dependency triple, and the word w_k is usually referred to as the modifier word. Some words (like verbs) carry more information than other words (such as a/an the) in one sentence. All relations strongly related to the "root word" produced by the Stanford dependency parser [16] are extracted. Figure 4 describes a procedure for extracting the dependency features. Consider two types of operations: find the root word and all relationships linked to the root word. If rel_i and w_j are root, then w_k will become a root word. We can take these relations as dependency features. If rel_i is not root and w_j or w_k as an equivalent root word, then these relations will become dependency features. Figure 3 shows how the method of dependency features is derived.

In the sentence, "*The fetus causes sickness discomfort and extreme pain to a woman during her pregnancy and labor.*", the root word is causes, and extracted relations which involve fetus, causes, discomfort, and pregnancy.

(1) det(fetus-2, The-1)	(9) case(woman-11, to-9)
(2) nsubj(**causes-3**, fetus-2)	(10) det(woman-11, a-10)
(3) root(ROOT-0, **causes-3**)	(11) nmod(discomfort-5, woman-11)
(4) compound(discomfort-5, sickness-4)	(12) case(pregnancy-14, during-12)
(5) dobj(**causes-3**, discomfort-5)	(13) nmod:poss(pregnancy-14, her-13)
(6) cc(discomfort-5, and-6)	(14) nmod(**causes-3**, pregnancy-14)
(7) compound(pain-8, extreme-7)	(15) cc(pregnancy-14, and-15)
(8) conj(discomfort-5, pain-8)	(16)conj(pregnancy-14, labor-16)

Fig. 3. Dependency parse

```
Algorithm 2: Dependency Features Extraction Method
Input   debateCorpus
Output  dependencyFeatures
for i size(debateCorpus) do
    sentences ← sentenceAnnotation(debateCorpus(i))
    for j in size(sentences) do
        dependencies ← dependencyAnnotation(sentences(j))
        for k in size(dependencies) do
            reln ← relation of dependencies(k)
            gov ← governor of dependencies(k)
            dep ← dependent of dependencies(k)
            if reln = root and gov = root then
                rootWord ← dep
                dependencyFeatures ← dependencies(k)
            end if
        end for
        for l in size(dependencies) do
            reln ← relation of dependencies(l)
            gov ← governor of dependencies(l)
            dep ← dependent of dependencies(l)
            if reln ≠ root then
                if gov = rootWord or dep = rootWord then
                    dependencyFeatures ← dependencies(l)
                end if
            end if
        end for
    end for
end for
```

Fig. 4. Dependency features extraction method

(3) Morphology Features

Morphology features concerns about the word’s part-of-speech. Using a transformation of dependency relation triples, the proposed system converts dependency features into "composite back-off features” [12] that generalize better than the traditional lexicalized dependency relationship features.

i. Head Back-off Features {rel_i, POS_j, w_k}, where the head word is replaced by its POS tag, but the modifier word is retained.
ii. Modifier Back-off Features {rel_i, w_j, POS_k}, where the modifier word is replaced by its POS tag, but the head word is retained.
iii. Full Back-off Features {rel_i, POS_j, POS_k}, where both the modifier word and the head word to their respective POS tags {POS_j and POS_k}.

Composite back-off features based on dependency relations, where only the head word is backed off to its POS tag, offer a useful alternative to encoding dependency relations as features for stance classification. On the accuracy metric, the head word backed off achieves a statistically significant improvement over the modifier word backed off features and fully backed off features. The proposed system used the head back-off features as morphology features.

3.4. Feature selection module

The selection of features aims to choose a small set of relevant features from the original ones by removing irrelevant, redundant, or noisy features. The proposed system used the Chi-Square method for feature selection. Selecting a (good) subset of features can give huge savings in computation time and increase accuracy. The proposed system retains all features with a score greater than zero.

$$X^2 = \sum_{i=1}^{m} \sum_{j=1}^{k} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \tag{1}$$

where,

- O_{ij} – Observed frequency
- E_{ij} – Expected frequency

For each feature, a contingency table is created with *m* rows and *k* columns. Each cell (*i*,*j*) denotes the number of rows having attribute feature as *i* and class label as *k*. For each feature in the dataset, the X^2 is calculated and then ordered in descending order according to the X^2 value. The higher the value of X^2, the more dependent the output label is on the feature and higher the importance the feature has on determining the output.

3.5. Stance classification module

This system used a Multinomial Naïve Bayes classifier [14] to evaluate the effectiveness of features for classifying debate posts. Naïve Bayes is a probabilistic classifier and based on Bayes rule. Naïve Bayes classifier works very well with text data and is fast in comparing to other algorithms. No optimization is required and performance is good. The classifier model is fast to build and implement easy to understand and easily updateable if new training data is received. For a post *p* and a class *c*,

$$P(c|p)=(P(p|c)P(c))/(P(p)) \tag{2}$$

For stance classification, accuracy defined in an equation (3) is used for evaluation.

$$Accuracy=(TP+TN)/(TP+FP+TN+FN) \tag{3}$$

- True positive (TP) is the number of posts that *support* the debate and is predicted to be a '*for*'.
- False positive (FP) represents the number of posts that *oppose* the debate and are predicted to be a '*for*'.
- True Negative (TN) is the number of posts that *oppose* the debate and are predicted to be an '*against*'.
- False Negative (FN) is the number of posts that *support* the debate and is predicted to be an '*against*'.

4. Experimental Setup

Four feature extraction models are compared by using three classifiers based on six ideological domains.

(1) Sentiment Analysis model (B1),

(2) Unigram model (B2),
(3) Dependency model (B3), and
(4) The proposed model (P1).

Sentiment Analysis Model (B1): The Stanford sentiment analysis tool used the first baseline model to obtain the sentiment labels at the sentence level. These labels are used as a feature to classify the stance of debate posts. There are 5 classes of sentiment classification: very negative (0), negative (1), neutral (2), positive (3), and very positive (4). For each post, a sentiment-to-stance assignment (mapping all 'positive' instances to 'for' and all 'negative' instances to 'against').

Unigram Model (B2): The second baseline model was used by the unigram features alone. The most common unigrams include very common words. Very common words are excluded from the unigrams.

Dependency Model (B3): The third baseline model was used by the set of dependency relations that is specific to a given parser. The Stanford parser is used for computing dependency relations. This model uses 37 grammatical relations.

Table 1. Classification results using Naïve Bayes Multinomial model

Feature Set	**Abortion**	**Creation**	**Gay Rights**	**God**	**Gun Rights**	**Healthcare**	**Average**
Baselines							
B1	66.65	51.62	66.28	69.27	67.86	67.63	64.88
B2	79.42	77.03	82.27	77.66	80.85	82.97	80.03
B3	87.02	86.89	90.74	87.64	88.14	85.16	87.60
Lexical features (L1)	57.60	56.03	50.78	57.70	52.20	52.50	54.47
Dependency features (D1)	70.49	70.76	73.01	71.25	68.30	68.28	70.35
Morphology features (M1)	71.63	74.71	75.00	71.04	71.02	71.56	72.49
P1(L1+D1+M1)	72.11	75.29	75.86	75.05	71.19	71.41	73.48
P2(P1+B2)	84.90	81.67	87.46	83.41	84.41	85.78	84.60
P3(P2+B3)	89.61	87.24	93.17	90.67	91.52	88.44	90.11

As seen in Table 1, baseline B3 that uses all relationships from the parse of dependency performs significantly better than other models that focus on selecting specific features. Once combined with one or more baseline models, the features we introduce (L1, D1, and M1) only become competitive. P3 is the highest performing model, combines the baselines of unigram (B2) and dependency (B3) with lexical features (L1), dependence features (D1), and morphology features (M1), and outperforms previously published approaches to the classification of stances mentioned in section 2 by a substantial margin. The features in section 3 of P1 are described earlier. P2, the first competitive model, adds P1 with unigram features. P3, which is the best performing classifier, also uses the features of arbitrary dependence. While we expected our baseline model B1 to perform poorly on this task using a sentiment classifier. The unigram model (B2) performed better than the sentence level sentiment tool (B1). This supports the findings of [2], which similarly found that features of sentiment did not prove helpful while features of unigram were difficult to beat. Additionally, we discover that the features of dependency (B3) have an even stronger baseline. The model of lexical feature (L1) is poorly performed because the coverage of (L1) in the dataset is less than 14 percentage of posts that contain at least one feature after the feature selected. Our best scoring system achieves an overall accuracy of 90.11%. Also, our model performs better for each of the topics explored during the debate. Our approach focuses on relationships of dependency which relate to stance words. Our results show it's helpful too. For evaluation, we compared our system to the following systems: Somasundaran and Wiebe used a) arguing-based features and b) sentiment-based features in [2]. Arguing-based features for each sentence in the post are positive or negative arguing speech by searching for trigram, bigram, and unigram matches with the argumenting lexicon. Modal words like "must" and "should" are typically strong markers of argumentation. Sentiment-based features are generated independently of arguing features. The sentiment polarity of the entire sentence and each content word in the sentence is calculated by using a sentiment lexicon. Mandya and his colleagues extracted a) topic stance features, b) stance bearing terminology, c) logical point features, d) unigram, and dependency features in [13].

Table 2. Comparison of the proposed system with previous work in terms of accuracy for six domains

Models	Accuracy					
	Abortion	**Creation**	**Gay Rights**	**God**	**Gun Rights**	**Healthcare**
Somasundaran and Wiebe [2]	60.55	63.96	63.71	-	70.59	-
Mandya and his colleagues [13]	89.40	87.99	90.18	88.05	93.51	-
Proposed model	89.61	87.24	93.17	90.67	91.52	88.44

Our results are directly compared to [2] and [13]. The experiment results were reported on the same dataset. First authors [2] got 60.55%, 63.96%, 63.71%, and 70.59% accuracy for Abortion, Creation, Gay Rights, and Gun Rights domains. The second authors [13] achieved a maximum of 89.40%, 87.99%, 90.18%, 88.05%, and 93.51% accuracy on five domains. Our best scoring reached an accuracy of 89.61%, 87.24%, 93.17%, 90.67%, 91.52%, and 88.44% in comparison to their accuracy on six domains. We additionally found that the accuracy of Abortion, Gay Rights, and God domains of our system are better than [13]. In [13], their system is the best accuracy on Creation and Gun Rights than other systems that are shown in Table 2. Surface and deep features that are useful for classifying long posts but lacking for short posts are extracted from the dependency parse. Debate posts in the Creation domain are short posts than other domains, therefore the accuracy of the proposed model for the Creation domain lower than that of [13]. The results of the evaluation are derived from 10-fold cross-validation and based on the same evaluation metrics, in particular, the accuracy depending on the number of debate posts. In this study, we extracted six popular issues from the monologic post dataset. These datasets are created by [2]. Table 3 shows the statistics of the dataset used. Each of these domains focuses on different topics allowing us to produce results over various domains.

Table 3. Statistics of the dataset

Datasets	**Abortion**	**Creation**	**Gay Rights**	**God**	**Gun Rights**	**Healthcare**
Original dataset	1151	1230	2064	953	1069	667
Somasundaran and Wiebe [2]	550	530	846	-	306	-
Mandya and his colleagues [13]	1030	856	1478	920	586	-
Proposed model	1040	862	1156	922	590	640

The proposed system used six different ideological domains and three different classifiers for training and testing using the 10-fold cross-validation technique. Logistic regression, SVM, and Naïve Bayes are used for stance classifiers. Among them, logistics is the worst performance and Naïve Bayes is the best performance. Observe that system using morphology features and dependency features performed significantly better than lexical features. Previous work suggests that for certain types of debates the baseline of unigrams can be hard to beat. The proposed feature extraction model outperforms the sentence-level sentiment model, while the situation is reversed for the unigram model and dependency model. The proposed feature analysis suggests that both surface and deep features are more insightful than sentiment features. The proposed method performs better than [2] and [13]. Previous research often approaches each target separately to define subjectivity expressed towards specific targets, ignoring the possible dependence that occurs between the targets, the related subjectivities, and other unknown factors involved. For example, the attitude toward one candidate in a social media election post may be strongly correlated with that toward another person, and such dependency exists widely in many other fields, including product reviews. In its complicated nature, the association of subjectivity could be correlated with concealed variables such as subjects under consideration (e.g., two political candidates on all issues are not actually against each other), among others. Lastly, we performed several experiments to investigate the advantages of modeling the interaction between stance labels. We have evaluated other possible systems, such as SVM classifiers and Logistic classifiers. The experiments demonstrated the Naïve Bayes' effectiveness in modeling a post.

5. Conclusion

Studying the classification of stance may be useful in defining political problems and recognizing how it influences popular attitudes. One of the latest tasks of opinion mining is the classification of the perspective of discussion: provided a post written for a subject of two-sided online debate, decide which of the two sides its author takes. Using a more advanced machine learning algorithm to have a high-quality stance classification system depends on how deep features are explored. For this model work, surface, and deep features explored and boosted the performance of the classifier. This method provides an automatic stance classification that is very useful for a variety of natural language analysis, such as rumour stance classification and fake news stance classification. The use of sarcasm and irony is one common way of expressing an opinion on social media. The system did not investigate the influence of the use of sarcasm and irony in opinion expression on the classification of stance in this study. The framework proposed decided to focus only on the text of content created by the user.

References

[1] S. Somasundaran and J. Wiebe, "Recognizing stances in online debates," in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - ACL-IJCNLP '09*, Suntec, Singapore, 2009, vol. 1, p. 226, doi: 10.3115/1687878.1687912.

[2] S. Somasundaran and J. Wiebe, "Recognizing Stances in Ideological On-Line Debates," in *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, 2010, pp. 116-124.

[3] P. Anand, M. Walker, R. Abbott, J. E. F. Tree, R. Bowmani, and M. Minor, "Cats Rule and Dogs Drool!: Classifying Stance in Online Debate," in *Proceedings of the 2nd workshop on computational approaches to subjectivity and sentiment analysis,* 2011, pp. 1-9.

[4] M. Walker, P. Anand, R. Abbott, and R. Grant, "Stance Classification using Dialogic Properties of Persuasion," in *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*, 2012, pp. 592-596.

[5] K. S. Hasan and V. Ng, "Why are You Taking this Stance? Identifying and Classifying Reasons in Ideological Debates," in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 2014, pp. 751–762, doi: 10.3115/v1/D14-1083.

[6] D. Sridhar, L. Getoor, and M. Walker, "Collective stance classification of posts in online debate forums," in *Proceedings of the Joint Workshop on Social Dynamics and Personal Attributes in Social Media*, 2014, pp. 109-117.

[7] A. Murakami and R. Raymond, "Support or Oppose? Classifying Positions in Online Debates from Reply Activities and Opinion Expressions," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, 2010, pp. 869-875.

[8] K. S. Hasan and V. Ng, "Extra-Linguistic Constraints on Stance Recognition in Ideological Debates," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2013, pp. 816-821.

[9] P. Sobhani, D. Inkpen, and S. Matwin, "From argumentation mining to stance classification," in *Proceedings of the 2nd Workshop on Argumentation Mining*, 2015, pp. 67-77.

[10] L. Wang, and C. Cardie, "Improving agreement and disagreement identification in online discussions with a socially-tuned sentiment lexicon," arXiv preprint arXiv:1606.05706, 2016.

[11] M.A. Walker, J.E.F. Tree, P. Anand, R. Abbott, and J. King, "A Corpus for Research on Deliberation and Debate," in *LREC*, 2012, Vol. 12, pp. 812-817.

[12] M. Joshi and C. Rosé, "Generalizing dependency features for opinion mining," in *Proceedings of the ACL-IJCNLP 2009 conference short papers*, 2009, pp. 313-316.

[13] A. Mandya, A. Siddharthan, and A. Wyner, "Scrutable Feature Sets for Stance Classification," in *Proceedings of the Third Workshop on Argument Mining (ArgMining2016),* Berlin, Germany, 2016, pp. 60–69, doi: 10.18653/v1/W16-2807.

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD explorations newsletter*, 2009, 11(1), pp.10–18.

[15] M.C. De Marneffe and C.D. Manning, "The stanford typed dependencies representation," in *Coling 2008: Proceedings of the workshop on Cross-Framework and CrossDomain Parser Evaluation*, pages 1–8.

[16] C.D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S.J. Bethard, and D. McClosky, "The Stanford CoreNLP Natural Language Processing Toolkit," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 2014, pp. 55-60.